JMLR: Workshop and Conference Proceedings vol 40:1–3, 2015

Open Problem: Learning Quantum Circuits with Queries

Jeremy Kun JKUN2@UIC.EDU
Lev Reyzin LREYZIN@MATH.UIC.EDU
Department of Mathematics, University of Illinois at Chicago

Abstract

We pose an open problem on the complexity of learning the behavior of a quantum circuit with value injection queries. We define the learning model for quantum circuits and give preliminary results. Using the test-path lemma of Angluin et al. (2009a), we show that new ideas are likely needed to tackle value injection queries for the quantum setting.

1. Introduction

Learning classical circuits with value injection queries (VIQ) was defined by Angluin et al. (2009b). In the VIQ model, one assigns values to subsets of the circuit wires and observes the output. This model has been extended to large alphabet, analog, and probabilistic circuits (Angluin et al., 2008, 2009a). There has also been work on learning networks with similar kinds of queries (Angluin et al., 2010; Kempe et al., 2003). Here, we ask how to learn a *quantum* circuit using VIQs. There is much work in physics on inferring the structure of specific quantum systems, but little on their generic learnability. Progress on this problem will bring together rich ideas from both the quantum and learning communities.

Model

A size k *quantum circuit* C on n qubits is a list of gates $G_1, \ldots, G_k$, where each G_i consists of an 8×8 unitary matrix $A_i : \mathbb{C}^8 \to \mathbb{C}^8$ and a list $(i_1, i_2, i_3) \in \{0, 1, \ldots, n\}^3$ of the indices of the qubits on which A_i operates. A *qubit* v is a unit vector in $\mathbb{C}^2$. Denote the basis vectors of $\mathbb{C}^2$ as e_0, e_1. The input to a quantum circuit is a unit vector in $(\mathbb{C}^2)^{\otimes n} = \mathbb{C}^{2^n}$, the n-fold tensor product of $\mathbb{C}^2$ with itself, where each copy of $\mathbb{C}^2$ corresponds to a qubit. We denote the basis vectors of $\mathbb{C}^{2^n}$ by e_i where $0 \leq i < 2^n$ is written as a binary string $i = b_1 \cdots b_s$ and $e_i = \bigotimes_{t=1}^{s} e_{b_t}$ is the tensor product of basis states from the copies of $\mathbb{C}^2$ (called a *pure* state). In our model all inputs to quantum circuits will be pure states.

The computation of an input vector $v = v_1 \in \mathbb{C}^{2^n}$ is as follows. In step i the following is applied to v_i: swap columns s.t. i_1, i_2, i_3 are the first 3 columns, apply the unitary map $A_i \otimes I_{2^{n-3}}$, and then reverse the column swap. Note that operations that only "depend" on three qubits can affect the entire state vector v_i. After all k gates are computed, v_{k+1} is measured, returning index i w.p. $|v_i|^2$, and v_{k+1} "collapses" to the basis vector e_i.

An *instance* of our model is a pair of integers n, k, an unknown quantum circuit C on n qubits of size k, and an unknown permutation σ on $\{1, \ldots, k\}$ masking the order of the gates. A *function injection query* (FIQ) is a tuple $(x, S, (B_i : i \in S))$ where $x \in \{0, 1\}^n$, $S \subset \{1, \ldots, k\}$, and B_i are 8×8 unitary matrices. The response to a query is a string y which is the final measured state of the quantum circuit formed by replacing the matrix A_i

	with measurements	observing full state vector
VIQ	**open** (fails test-path lemma)	**open**
FIQ	poly	poly

Table 1: Summary of our results for learning quantum circuits with open problems in bold.

of gate G_i with $B_{\sigma(i)}$ for each $i \in S$ and run on the starting state vector e_x. We define an *unmeasured* variant where the query response is the entire state vector before measurement.

Given $v \in \{0,1\}^3$, a *value injection* of v into a gate G_j of a circuit C with matrices A_j operating on bits (i_1, i_2, i_3) is an augmented circuit C' with three additional qubits fixed to the states $e_{v_1}, e_{v_2}, e_{v_3}$ and an additional gate H_j following G_j defined by swapping the new qubits with i_1, i_2, i_3, respectively. One can similarly inject values into a subset of gate outputs. A *value injection query* (VIQ) is a tuple $(x, S, (v_i \in \{0,1\}^3 : i \in S))$ where $x \in \{0,1\}^n$ and $S \subset \{1, \ldots, k\}$. The response to a query is a string y which is the measurement of the first n qubits of the final state of the quantum circuit formed by injecting each value v_i into gate $\sigma(i)$ of C and running the resulting augmented circuit on input e_x. We define the analogous unmeasured variant as well.

An algorithm *learns to ε-behavioral equivalence* if for any circuit C it outputs C' s.t. the distributions over responses to VIQs/FIQs for C and C' differ by $\leq \varepsilon$ in l_1 norm. The *query complexity* of an algorithm is measured in the total number of VIQs/FIQs it uses, and it has been extensively studied for classical circuit families (Angluin et al., 2008, 2009a,b).

Open Problem *Determine the query complexity of learning quantum circuits with VIQs.*

2. Preliminary Results

Now we give some preliminary results on learning quantum circuits with VIQs and FIQs. These results elucidate which of the questions in Table 1 are known and which remain open.

2.1. Learning quantum circuits with FIQs

Proposition 1 *There is a $O(k \log k + kn)$-query algorithm for learning a quantum circuit with FIQs (without measurement) to 0-behavioral equivalence.*

Proof The algorithm learns the gates individually and then their order. First, for each gate $1 \leq j \leq k$ and $0 \leq i \leq 7$, query $Q_{j,i} = (i0^{n-3}, [k] \setminus \{j\}, (I_8)_{l \in S})$. With $O(n)$ overhead (see below), assume gate G_j acts on the first 3 qubits. Fixing j and looking at all $Q_{j,i}$, each output vector consists of the ith column of A_j. Repeat on all gates makes $O(nk)$ FIQs. To find the qubits G_j acts on: inject the identity everywhere except j and a 3-cycle permutation at j, and query each e_{1_l} where 1_l is the string of 0s with a 1 at position l.

Next sort the gates in $O(k \log k)$ queries, ordering two gates $G_j, G_{j'}$ as follows. Fix two permutations $\sigma, \sigma' \in S_8$ that do not commute. Form B_σ by permuting the columns of the identity matrix according to σ. Inject the identity into every gate except j, j', B_σ into j, and $B_{\sigma'}$ into j'. Choose x such that $\sigma\sigma'(x) \neq \sigma'\sigma(x)$ and input e_x. If the order of j, j' matters the response to the query will differ, and it is trivial to determine the correct order. Note the comparisons do not need the full state vector. The query complexity is $O(nk + k \log k)$. ■

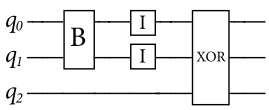
Measurement adds poly(k/ε) FIQs to the above algorithm to make it ε-approximate. Consider the case with two gates $G_j, G_{j'}$ with $A_j, A_{j'}$ operating on the same qubits. Decompose A_j according to a basis of the vector space $SU(8)$. Inject the identity everywhere but j, j' and inject all pairwise sums of basis operations to $A_{j'}$, query all 8 basis vectors, and measure the outputs. Using the basis coefficients of A_j as variables, this gives a system of polynomial equalities in which all parameters are $O(1)$. It can be solved to accuracy ε/k. It is easy to see that the errors across gates grow linearly, and the general case is similar.

2.2. Quantum circuits fail the test path lemma with measured VIQs

Angluin et al. (2009a) use the idea of "test paths" (from Angluin et al. (2009b)), to bound the query complexity of learning boolean probabilistic circuits. Their analysis also shows test paths fail for probabilistic circuits with an alphabet size greater than two. We show quantum circuits have the same barrier by constructing a gadget analogous to that of Angluin et al. (2009a) (Lemma 8). Define B as a matrix as in Figure 1. Further define the standard quantum XOR gate using an extra scratchwork qubit by the mapping $e_{ijk} \mapsto e_{ij(k\oplus i\oplus j)}$.

Lemma 2 *There is a circuit on which every (measured) VIQ leaving a path free makes the last output qubit uniformly random, yet with no VIQ the last output qubit is deterministic.*

Proof Define C on three qubits as in Figure 1. The circuit normally maps $e_{000} \mapsto \frac{1}{\sqrt{2}}(e_{000} + e_{110}), e_{010} \mapsto \frac{1}{\sqrt{2}}(e_{000} - e_{110}), e_{100} \mapsto \frac{1}{\sqrt{2}}(e_{111} + e_{101})$, and $e_{110} \mapsto \frac{1}{\sqrt{2}}(e_{111} - e_{101})$, i.e., the last qubit is 1 iff the input's first qubit is 1. Likewise, when the scratch-work qubit is 1, the output bit is flipped. When there is a VIQ, say, of 1 at the identity gate acting on the 2nd qubit, the mapping becomes $e_{000} \mapsto \frac{1}{\sqrt{2}}(e_{0110} + e_{1101}), e_{010} \mapsto \frac{1}{\sqrt{2}}(e_{0110} - e_{1101}), e_{100} \mapsto \frac{1}{\sqrt{2}}(e_{0111} + e_{1100})$, and $e_{110} \mapsto \frac{1}{\sqrt{2}}(e_{0111} - e_{1100})$. The extra qubit introduced by the VIQ is the last index, and the qubit of interest is the third qubit, which is uniformly random. ■



(a)

(b) $B = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \end{pmatrix}$

Figure 1: Left: the circuit for Lemma 2. Right: the columns of B form the Bell basis.

References

Dana Angluin, James Aspnes, Jiang Chen, and Lev Reyzin. Learning large-alphabet and analog circuits with value injection queries. *Mach. Learn.*, 72(1-2):113–138, 2008.

Dana Angluin, James Aspnes, Jiang Chen, David Eisenstat, and Lev Reyzin. Learning acyclic probabilistic circuits using test paths. *J. Mach. Learn. Res.*, 10(Aug):1881–1911, August 2009a.

Dana Angluin, James Aspnes, Jiang Chen, and Yinghua Wu. Learning a circuit by injecting values. *J. Comput. System Sci.*, 75(1):60–77, January 2009b.

Dana Angluin, James Aspnes, and Lev Reyzin. Optimally learning social networks with activations and suppressions. *Theor. Comput. Sci.*, 411(29-30):2729–2740, 2010.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD*, pages 137–146, 2003.